

Sub
A2

What Is Claimed Is:

1 Claim 1. A method of transferring between types of conversion
2 processes in a computer which converts instructions from a target
3 instruction set to a host instruction set comprising the steps of:

4 executing code morphing software including an interpreter and a
5 translator to generate host instructions from target instructions,
6 detecting at intervals whether the interpreter or the translator is
7 operating,

8 increasing a count if the interpreter is operating and decreasing the
9 count if the translator is operating, and

10 changing from interpreting to translating a sequence of target
11 instructions when the count reaches a selected maximum.

1 Claim 2 A method as claimed in Claim 1 in which the interval is a
2 selected time period

1 Claim 3. A method as claimed in Claim 1 in which the interval is a
2 selected number of executed target instructions.

1 Claim 4. A method as claimed in Claim 1 in which the amount the
2 count is increased at a detection of interpretation is selectable.

1 Claim 5. A method as claimed in Claim 1 in which the amount the
2 count is decreased at a detection of translation is selectable.

1 Claim 6. A method as claimed in Claim 1 comprising the further steps
2 of:

3 counting each instance in which a sequence of instructions is
4 interpreted,
5 changing from interpreting to translating a sequence of target
6 instructions when the count of instances reaches a selected maximum.

1 Claim 7. A method as claimed in Claim 7 comprising the further steps
2 of:

3 gathering statistics regarding each sequence of instructions, and
4 optimizing translation of a sequence of instructions based on statistics
5 gathered.

1 Claim 8. A method as claimed in Claim 1 in which the step of
2 changing from interpreting to translating a sequence of target
3 instructions when the count reaches a selected maximum includes
4 translation with limited optimization, and

5 which further includes the steps of:

6 testing while executing a sequence of target instructions translated with
7 limited optimization to determine whether the sequence should be
8 further optimized, and
9 retranslating and further optimizing in response to the testing.

1 Claim 9. A method as claimed in Claim 8 in which the step of testing
2 while executing a sequence of target instructions translated with limited
3 optimization includes counting each instance in which a sequence of
4 instructions is executed, and

Cont
A2
5 the step of retranslating and further optimizing occurs when the count of
6 instances reaches a selected maximum.

1 Claim 10. A method of optimizing execution by a computer which
2 dynamically converts instructions from a target instruction set to a host
3 instruction set comprising the steps of:

4 providing a plurality of instruction conversion processes each providing a
5 different level of optimization for converted instructions from a target
6 instruction set to a host instruction set,

7 providing means for determining dynamically which conversion process
8 best converts each sequence of instructions, and

9 converting a sequence of instructions using a conversion process
10 determined to best convert the sequence of instructions.

1 Claim 11. A method as claimed in Claim 10 in which the conversion
2 processes include interpretation and translation.

1 Claim 12. A method as claimed in Claim 10 in which the conversion
2 processes include interpretation, translation with minimal optimization,
3 and translation with advanced optimization..

1 Claim 13. A method as claimed in Claim 10 in which the means for
2 determining dynamically which conversion process best converts each
3 sequence of instructions depends on the number of times each sequence
4 is converted by a particular conversion process.

1 Claim 14. A method as claimed in Claim 10 in which the means for
2 determining dynamically which conversion process best converts each

Cont
A2

3 sequence of instructions depends on a ratio of the number of times one
4 conversion process is run compared to another conversion process.

1 Claim 15. A method as claimed in Claim 10 in which the means for
2 determining dynamically which conversion process best converts each
3 sequence of instructions

4 depends on the number of times each sequence is converted by a
5 particular conversion process, and

6 depends on a ratio of the number of times one conversion process
7 is run compared to another conversion process.

Add
A2

Add
B1

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50